Amendments to the Specification

Please replace the paragraph beginning on page 8, line 3, with the following amended paragraph:

Accordingly, the present invention provides a method for dynamic task assignment in a workflow. In one embodiment, the present invention provides a method of assigning resources to nodes in workflow. In the present embodiment, the method is comprised of defining a plurality of nodes where each of the nodes are tasks to be executed. Further, the method of the present embodiment is further comprised of defining the resources authorized for the execution of each of the nodes. Additionally, the method of the present embodiment is further comprised of storing a set of data items which include variables pertaining to the workflow execution in the workflow. Also, the method of the present embodiment is further comprised of assigning the defined resources to the nodes for the execution thereof in accordance with a set of rules which are utilized to control the execution of the workflow. Pre-computed authorizations can be applied to the resources and wherein the authorizations are maintained in an up-to-date-manner, such that a set of authorized resources is retrievable concurrent with an initiated task.

Please replace the paragraph beginning on page 22, line 12, with the following amended paragraph (Please Note: "level hierarchy" was originally underlined in this paragraph and is not being added to this paragraph):

Analogously, ~~Figure 3B~~ Figure 5B is an example of level hierarchy for workflow 300 of Figure 3. A level hierarchy $LH \subseteq L \times L$, is defined for levels which is a partial order relation $\leq_L$ on L. Given l, l' $\in$ L, l' holds if l precedes l' in the order. The graphical notation for the level hierarchy follows the same conventions of the role hierarchy as described in Figure 5A. Accordingly, authorizations applied to secretary 509 (descendent) are inherited by medical consultant 507 (parent). It also follows that senior manager 503 inherits authorizations applied to junior manager 505 and medical consultant 507, and because secretary 509 is a descendent of medical consultant 507, senior manager 503 also inherits authorizations applied to secretary 509.

Please replace Table 3 beginning on page 33, line 3, with the following amended Table 3:

LevelHierarchy	Level	Parent
	Secretary	~~Junior Manager~~ Medical Consultant
	Junior Manager	Senior Manager

Medical Consultant   Senior Manager

Please replace the paragraph beginning on page 33, line 8, with the following amended paragraph (Please Note: "R-Play" and "L-Play" were originally underlined in this paragraph and are not being added to this paragraph):

Relations R-Play and L-Play define play authorizations. These relations are populated as the organization schema is defined; relation R-Play is also typically updated as new workflow schema is defined. In workflow ~~system 100~~ system 300, the R-Play relations are shown in Table 4 and the L-Play relations are shown in Table 5 as follows:

Please replace the paragraph beginning on page 33, line 29, and ending on page 34, line 3, with the following amended paragraph (Please Note: "R-Execute", "L-Execute" and "Type" were originally underlined in this paragraph and are not being added to this paragraph):

Relations R-Execute and L-Execute defines execute authorizations. They are populated as a new workflow schema is defined, and can be later modified at the occurrence of specified events on time or workflow data. Attribute Type defines whether the authorization is explicitly specified for the role/level or if it has been derived by means of derivation rules. With reference to the workflow ~~100 of Figure 1~~ 300 of Figure 3, Table 6, below, shows the R-Execute relations and Table 7, below, shows the L-Execute relations.

Please replace the paragraph beginning on page 39, line 4, with the following amended paragraph (Please Note: "binding of duties", "separation of duties" and "restricted task execution" were originally underlined in this paragraph and are not being added to this paragraph):

As described above, active rules are used as a suitable paradigm for modeling and enforcing authorization constraints. Further, examples of such rules, for the above described different constraint categories are described below. Rule examples are defined for the workflow schema 300 shown in ~~Figure 31~~ Figure 3. In particular, examples of constraints, e.g., binding of duties, separation of duties, and restricted task execution are provided.

Please replace the paragraph beginning on page 40, line 27, and ending on page 41, line 2, with the following amended paragraph (Please Note: "secretary" and "filing" were originally underlined in this paragraph and are not being added to this paragraph):

The first rule grants the permission to all agents playing role <u>secretary</u> to start executing task <u>filing</u> every day at 8 a.m. The second rule revokes the same permission every day at 6 p.m. It should be appreciated that, unlike the previous example, these rules define a case-independent constraint, which holds for every instance of the Medical Insurance schema in ~~workflow 100 of Figure 1~~ <u>workflow 300 of Figure 3</u>.